The Effect of Initial Slow-start Threshold Size in DCCP over Large Delay Link Networks

Shahrudin Awang Nor, Suhaidi Hassan SMIEEE, Omar Almomani
InterNetworks Research Group
Graduate Department of Computer Science
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok
Kedah, MALAYSIA
{shah, suhaidi }@uum.edu.my, s90637@student.uum.edu.my

Abstract

As an alternative to UDP, DCCP is gaining popularity as a new transport protocol for sending streaming multimedia contents in the Internet nowadays. DCCP is a new unreliable transport protocol which has built-in congestion control unlike UDP. With this feature, DCCP ensures that there is no bandwidth monopoly by certain transport protocol like UDP did for years. UDP has been proven that can eat up all the available bandwidth in the Internet while competing with other transport protocols such as TCP in carrying streaming audio and multimedia traffic in many situations. In this paper, we show that initial slow-start threshold (ssthresh) size has a significant effect to the performance of DCCP carrying VoIP traffic over large delay link. Slow-start threshold is used by congestion control implemented in DCCP's TCP-like congestion control (CCID2) where all TCP congestion control implementations are required to support it. From our experiments, we found out that too small initial slow-start threshold value for large delay link makes the traffic throughput sent using DCCP requires longer time to become stable. The selection of suitable value of initial slow-start threshold is then vital to the stability of the network carrying streaming audio data over DCCP.

Keywords: *DCCP, slow-start threshold, TCP-like*

1. INTRODUCTION

UDP (User Datagram Protocol) has been used as a transport protocol for years before the emergence of DCCP (Datagram Congestion Control Protocol). As UDP is an unreliable transport protocol, it is suitable to be used for the delivery of streaming multimedia data over the Internet. Moreover, UDP has simpler packet header with no built-in congestion control and the sender can send the UDP packets as much as possible without knowing what is happening in the entire network. There is no packet retransmission using UDP because it is an unreliable transport protocol that will not detect any packet loss. To certain extend, packet loss is acceptable in UDP as far as the quality of audio or video of streaming media can be accepted by receivers.

The main problem with the usage of uncontrolled UDP as transport protocol is the congestion in the network due to too much UDP packets sending by the sender. As the senders does not know the congestion that may happen in the network, or their UDP packets will dominate the entire bandwidth, other reliable transport protocol such as TCP (Transmission Control Protocol) well be punished with less or no bandwidth left to be used to send reliable data. TCP is well known as a reliable transport protocol with larger packet header. Congestion control in TCP can prevent congestion in the network efficiently and it is widely deployed in the Internet nowadays.

Initial slow-start threshold size in DCCP affects the stability of throughput. There are two standards for congestion control mechanism scheme for DCCP: Congestion Control Identification 2 (CCID2) [3] and

978-1-4244-2315-6/08/$25.00 ©2008 IEEE.

Congestion Control Identification 3 (CCID3) [4]. CCID2 and CCID3 use TCP-like congestion control and TCP Friendly Rate Control (TFRC) [6] mechanism, respectively. TCP-like congestion control is similar to the congestion control mechanism deployed by TCP. As an addition, there is an experimental Congestion Control Identification 4 (CCID4) [5] for Small-Packet (SP) variant of TFRC.

In this case, we only concentrate on CCID2 which is TCP-like congestion control because slow-start threshold only involves in congestion control which is similar to congestion control of TCP. CCID2 congestion control can be used over large delay link whereas CCID3 is not very suitable for such link because the behaviour of TFRC that does not sense the network link actively to adjust the sending rate. CCID2 can handle abrupt changes in network throughput rate whereas CCID3 is more suitable for link with more stable network link.

In this paper, we investigate the effect of having different size of initial slow-start threshold value for DCCP on the stability of bottleneck throughput. The rest of the paper is organized as follows; Section 2 discusses the introduction of initial slow-start and threshold in DCCP; Section 3 is about the CCID2 congestion control for DCCP; Section 4 explains the experimental design, Section 5 details all the simulation results, Section 6 gives the simulation analysis, and Section 7 concludes the paper.

2. INITIAL SLOW-START AND THRESHOLD SIZE

In DCCP, initial slow-start threshold value in a unit of number of packets applies on CCID2 only because it is adopted from the slow-start threshold for conventional TCP congestion control. CCID2 can sense the network changes more actively unlike CCID3 which uses TFRC as a congestion control mechanism.

Slow-start algorithm [9] for congestion control uses a third parameter called threshold, initially 64K, in addition to the receiver window and congestion window. According to this algorithm, when timeout occurs, the threshold is set to half of the current congestion window, and the congestion window is reset to one maximum segment. Slow-start is the used to decide what the network can handle, apart from that exponential growth stops when the threshold is strike. From that point on, known as congestion avoidance phase, thriving transmissions grow the congestion window linearly by one maximum segment for every burst or Round-Trip Time (RTT) instead of one per segment. Although the strategy is referred to as "slow-start", its congestion window growth is quite aggressive.

There is a variation to the slow-start algorithm known as fast recovery. In the fast recovery algorithm, during congestion avoidance mode, when packets detected through 3 duplicate ACKs are not received, the congestion window size is reduced to the slow-start threshold, rather than the smaller initial value.

One of the problem that may arise is that slow-start assumes that unacknowledged segments are due to network congestion. While this is an acceptable assumption for a lot of networks, segments may be lost for other reasons, such as poor data link layer transmission quality. Thus, slow-start can execute badly in situations with poor reception, such as wireless networks.

3. DCCP CCID2 (TCP-LIKE)

Datagram Congestion Control Protocol [10] as defined by IETF is well suited as a transport protocol for delivering multimedia traffic over wired or wireless networks. It supports bidirectional unicast connections of congestion-controlled unreliable datagram. DCCP is the right choice for applications that used to transfer huge amounts of data such as streaming multimedia traffic that can take advantage from control over the tradeoff between timeliness and reliability. It is also good for network health due to its built-in congestion control features.

UDP [12] is a connectionless transport protocol which has been a popular protocol for sending multimedia data in the Internet and is used widely by many applications. Although UDP can avoid long delays and works well with the delivery the multimedia data, but it is likely a threat for future networks. Nowadays, more and more multimedia applications are increasingly used and as a result it can collapse the entire network if not controlled properly. This is because UDP provides no congestion control at all, even its applications can have congestion control at higher layer implemented on their own, but it will add additional works or burden to application programmers. On the other hand, if DCCP is used, congestion control is already included and provided at transport layer protocol, so programmers can concentrate and focus solely on the applications and not the lower layer congestion control provided by transport protocol. As a result, this will be an advantage to the development of applications in term of time consumed for developing applications.

TCP [8] is not well suited for streaming media due to its reliable in-order delivery and congestion control

that can cause randomly long delays. Its reliable in-order delivery mechanism has to retransmit the packet if there is a packet loss happen during transmission in the network. Packet loss in TCP is detected either by time out or three duplicate acknowledgements received by sender from the network. For this reason, if the transmission of streaming media is affected so much by delays, it can disrupt the quality of service received by end users. Hence, TCP is only suitable for applications that rely on the reliability and can tolerate the delay like traditional web applications, file transfer and email.

Most real-time multimedia traffic ranging from interactive applications such as Voice over Internet Protocol (VoIP) and video conferencing to non-interactive applications like audio and video streaming are commonly use unreliable transport protocol UDP as their transport protocol. UDP provides best transport platform to deliver error-tolerant and delay-intolerant traffic. This is due to the some features of UDP like simpler connectionless implementation, shorter packet header, no congestion control, no acknowledgement, no retransmission, and etc. Even UDP can serve real-time multimedia traffic very well; there is a friendliness issue with other transport protocol like TCP which delivers reliable best-effort service for error-intolerant and delay-tolerant data like World Wide Web, email, file transport, and etc. In competing with TCP traffic when there is bandwidth restriction, UDP traffic will consume more and will dominate all the bandwidth while TCP traffic is halted. The same thing can happen when competing with other lower bandwidth applications, such as wireless link, and as a result real-time traffic utilizing UDP will consume 100% of bandwidth link utilization.

4. EXPERIMENTAL DESIGN

The experiments have been carried out by means of simulation with the simulation topology as shown in Figure 1. There is one TCP sender sending File Transfer Protocol (FTP) data to TCP receiver as background traffic. DCCP sender 1, sender 2 and sender 3 send VoIP data to DCCP receiver 1, receiver 2 and receiver 3, respectively. To avoid packet drop, Sender 1, sender 2 and sender 3 were not started sending VoIP data simultaneously. There is a delay of 200 milliseconds for the start time of every sender. Router 1 and router 2 were connected using 2 Mbps bottleneck link. For simplicity, instead of using other type of queue management such as Random Early Detection (RED), the type of queue management used in this link was DropTail, which implements First-In First-Out (FIFO). The network simulator ns-2 [1] with DCCP module [11] installed was chosen to simulate the DCCP CCID2 performance using different size initial slow-start threshold value.

The initial slow-start threshold value used was in term of the number of packets. In addition, the VoIP codec used was G.711 which has the attributes of 64 kbps of transfer rate, 50 packets per second, packet size of 160 bytes and 20 ms delay. Background traffic in all the experiments was simulated as bulk TCP flows with infinite FTP sources. The throughput was measured between Router 1 and Router 2, which is a bottleneck link in the simulation topology.

The network simulation topology used was the classic dumb-bell topology. Dumb-bell topology is a very common topology that has been used in many TCP network simulations. For the router to router connection, the large delay bottleneck link had been set to have a bandwidth of 2 Mbps with 300 ms propagation delay. This large delay bottleneck link can be used as an emulation of satellite or wireless links with a fixed forward link delay of 300 ms and fixed return link delay of 300 ms. This assumption is reasonable based on Henderson et al. [7] for satellite link. There is also research done by other researchers that use this assumption for large delay link [13]. We also considered that the bottleneck link has enough bandwidth allocation for the data transfer to flow from the sender to the receiver.

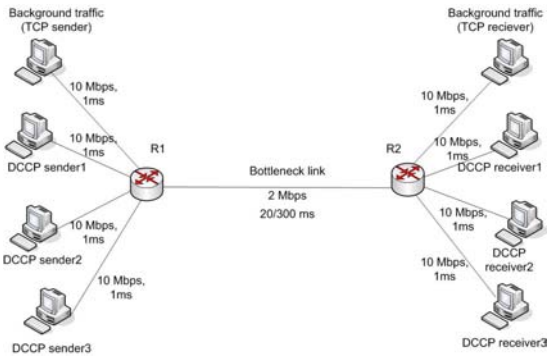

Figure 1. Simulation topology

4.1 Experiment 1:

This preliminary experiment is to show how UDP and DCCP flows can coexist with TCP flow within same bottleneck link and sharing the same bandwidth. In this first experiment, the TCP sender sent infinite data using FTP and the other two DCCP senders sent CBR data with 800kbps data rate. In the first part, TCP flow was started first for 20 seconds and then followed by two UDP flows. We assumed that 20 seconds is enough for the TCP flow to utilize the bandwidth without any contention with other flows,

so we can see the effect on throughput of having other flows joining the bottleneck link after that. For the second part, the TCP flow was simulated with two DCCP flows in the same way as the first one. DCCP flows with CCID2 and CCID3 congestion controls were used in the second part of this experiment.

4.2 Experiment 2:

As mentioned earlier, we are concentrating on DCCP with CCID2 congestion control for delivering VoIP traffic over large delay link. A simulation to compare the performance of DCCP with CCID2 and with CCID3 was carried out. For DCCP with CCID2 only, a simulation was also carried out to see the effect of having different initial slow-start threshold values for 20 ms and 300 ms bottleneck link. The chosen initial slow-start threshold values were 20, 50, 100 and 200 packets. In this experiment, the DCCP senders sent VoIP data with G.711 codec which has the attributes of 64 kbps of transfer rate, 50 packets per second, packet size of 160 bytes and 20 ms delay. The 20 ms and 300 ms delay links were used to represent the small and large delay links.

4.3 Experiment 3:

With large delay of 300 ms for the bottleneck link, the simulation was done with DCCP flows having different value of initial slow-start threshold. The DCCP CCID2 flows started after 20 seconds with background traffic of FTP sending infinite data and sharing the same bandwidth. The VoIP codec used was the same, i.e. G.711. In this experiment, it is anticipated to see how the different initial values of slow-start threshold affect the throughput of DCCP flows when they coexist together with other background flow, specifically in this case the TCP flow.

5. SIMULATION RESULTS AND ANALYSIS

As an additional feature, ns-2 provides a useful tool called Network Animator (NAM) to be used to view results by means of graphical display, as shown in Figure 2. All significant events during a simulation can be easily monitored. Nam is a Tcl/TK based animation tool for viewing network simulation traces and real world packet tracedata [2]. During an ns-2 simulation, nam trace file which has topology configurations, layout information, and packet traces can be generated using tracing events in ns-2.

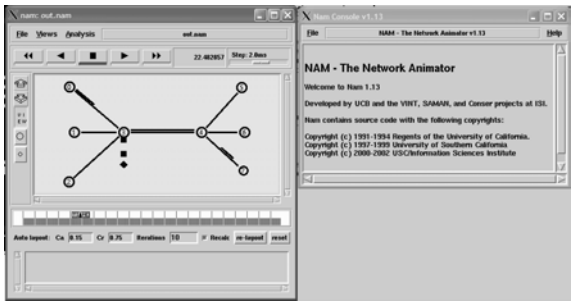

Figure 3. The NAM outlook

5.1 Experiment 1:

We first present the result that shows how UDP dominates the entire bandwidth when coexist with TCP in limited bandwidth bottleneck link. In the first 20 seconds as shown in Figure 4, TCP flow gets enough bandwidth in the link. The FTP can not utilize the entire bandwidth after it just started because of the large delay introduced by the link. It is seen clearly that after 20 seconds, the bandwidth is dominated almost entirely by UDP flows after they were started. Notably, UDP ate up all the bandwidth and left none for TCP.

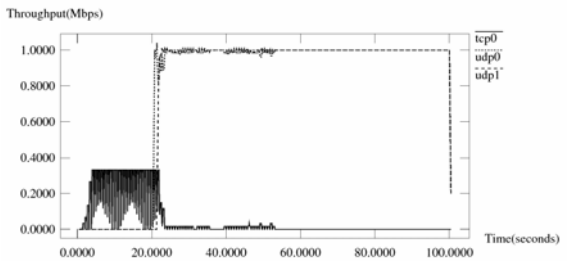

Figure 4. Single TCP flow with 2 UDP flows (300 ms bottleneck link)

It was proven in this preliminary experiment that UDP aggressively will dominate the entire bandwidth if coexist with TCP. This will lead to the lack of available bandwidth for TCP and the TCP flow will completely down, and to some extend, the network will collapse due to this. On the other hand, DCCP is friendlier when coexist with TCP due to the characteristics of DCCP which is more compromising with other flows.

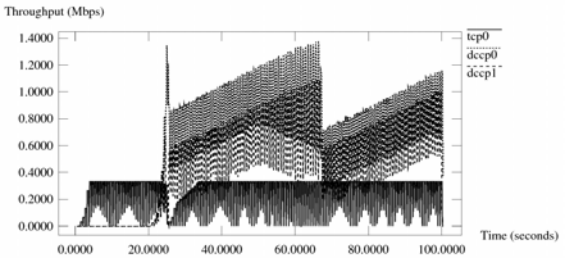

Figure 5. DCCP using CCID2 congestion control (300 ms bottleneck link)

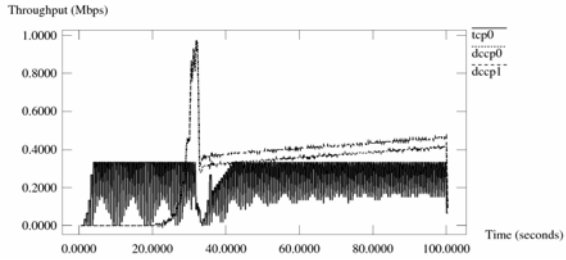

Figure 6. DCCP using CCID3 congestion control (300 ms bottleneck link)

Experiment results as shown in Figure 5 and Figure 6 show that how DCCP flows can compromise with TCP flow. It is noticed that TCP throughput maintains even after DCCP flows started at time 20 seconds. Despite the fact that there is a temporary little drop in TCP throughput when the DCCP flows just started, and consequently after a while it gains its throughput. It is also shown from Figure 5 and Figure 6 that the throughput for DCCP flow using CCID2 with TCP-like congestion control when delivering VoIP packet is relatively higher than DCCP flow using CCID3 with TFRC congestion control for large delay link.

5.2 Experiment 2:

The result in Figure 7 shows that for a large delay link, in this case 300 ms link, DCCP flow with CCID3 congestion control suffers from bandwidth. Here, in contrast with Experiment 1, DCCP with CCID3 senders transmitting a low rate CBR traffic, i.e. VoIP data with 64 kbps over the large delay link. It is shown that DCCP flow with CCID2 congestion control can ramp up to the encoding rate of the application when sending VoIP data in this link whereas DCCP flow with CCID3 congestion control performs badly and get very low throughput.

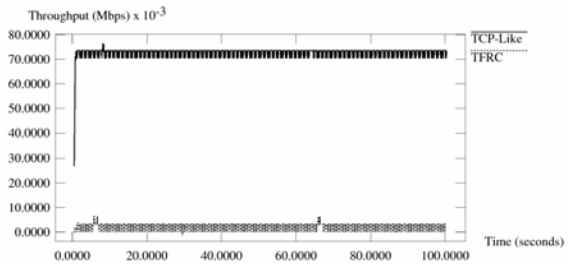

Figure 7. DCCP with CCID2 (TCP-like) and CCID3 (TFRC) congestion control (300 ms bottleneck link)

The inferior performance of CCID3 when sending small packet, in this case VoIP packet of 160 bytes, leads to the proposal of a new CCID4 TFRC with small packet (TFRC-SP) by IETF which is still under active investigation and not yet finalized.

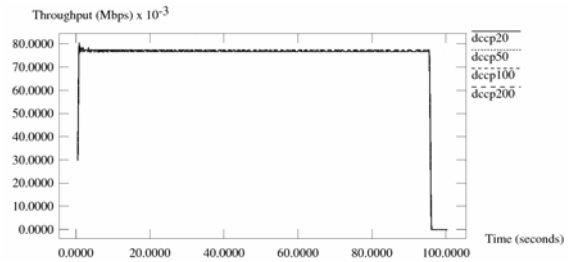

Figure 8. DCCP CCID2 throughput for 20 ms bottleneck link

The result in Figure 8 shows that the throughput for DCCP CCID2 using variety values of initial slow-start threshold does not give much difference in carrying VoIP traffic over the low delay link. All the DCCP flows with different values of initial slow-start threshold are having smooth throughput from initial starting point until the end of the simulation time.

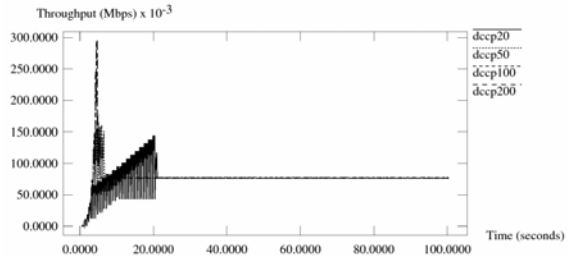

Figure 9. DCCP CCID2 throughput for 300 ms bottleneck link

For the simulation using large delay link as shown in Figure 9, there is a significant difference between DCCP CCID2 flows using different values of initial

slow-start threshold. It is seen that CCID flow with least initial slow-start threshold value, i.e. 20 packets requires longer time to get to its stable throughput. It is also shown in the graph that CCID flow with initial slow-start threshold value of 200 packets is the fastest flow to get stable throughput.

5.3 Experiment 3:

In all the cases, there is no packet drop. Figure 10 shows the throughput of all DCCP CCID2 flows using different values of initial slow-start threshold, i.e. 20, 50, 100, 150 and 200 packets. It can be seen that DCCP CCID2 flow with initial slow-start threshold value of 200 packets gives the fastest time to become stable whereas flow with the value of 20 packets takes the longest time. It is obvious that the DCCP flows with smaller initial slow-start threshold values give high sharp start of throughput for a short time and then the throughput were adjusted quickly to the application encoding.

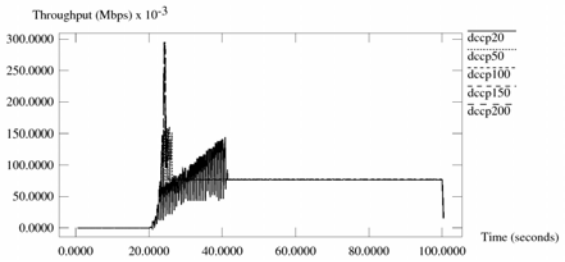

Figure 10. DCCP CCID2 flows with different initial slow-start threshold values

For better view, the graphs of DCCP CCID2 flows with initial slow-start threshold value of 20, 50, 100, 150 and 200 packets are shown separately in Figure 11, Figure 12, Figure 13, Figure 14 and Figure 15 respectively.

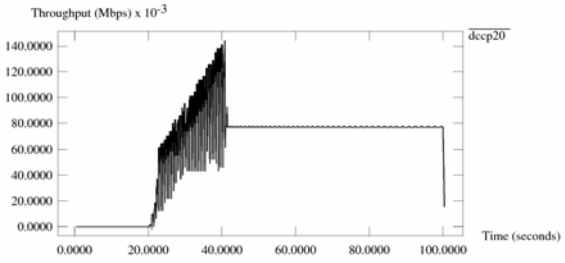

Figure 11. DCCP CCID2 with initial slow-start threshold value of 20 packets

In the simulation experiment of DCCP CCID2 with initial slow-start threshold value of 20 packets, the end-to-end delay was measured. The end-to-end delay from time of 20 seconds to 40 seconds is 1.154 milliseconds whereas 1.151 milliseconds is the end-to-end delay from time of 40 seconds to 100 seconds. Even though the difference is quite small, this shows that higher end-to-end delay is introduced during unstable throughput period from time 20 seconds to 40 seconds. We assume this period as unstable because we can see that during this period, the throughput fluctuated very much and some of the throughput values shown are far below the VoIP G.711 encoding rate which is 64 kbps. Moreover, VoIP is an interactive real-time streaming and it requires a stable throughput to perform well. As a result, this will badly affect the perceived quality of voice delivered to the end users.

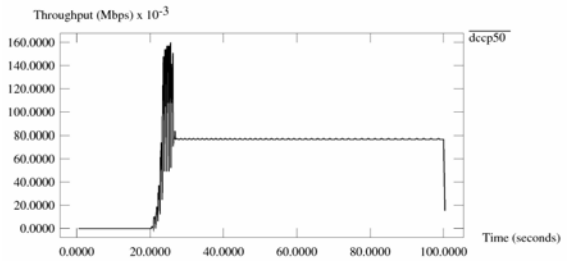

Figure 12. DCCP CCID2 with initial slow-start threshold value of 50 packets

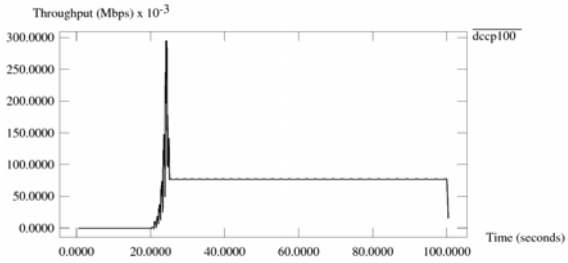

Figure 13. DCCP CCID2 with initial slow-start threshold value of 100 packets

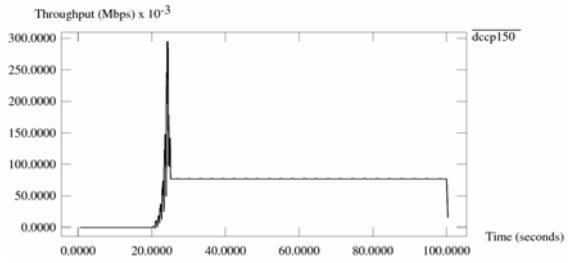

Figure 14. DCCP CCID2 with initial slow-start threshold value of 150 packets

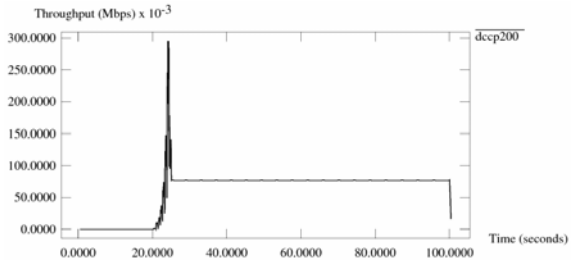

Figure 15. DCCP CCID2 with initial slow-start threshold value of 200 packets

Initial slow-start threshold value affects the performance of DCCP using CCID2 congestion control. For link with delay of 300 ms, longer time is needed if slow-start threshold value is very small for the throughput to become stable. If higher slow-start threshold value is used, the time for the throughput to stabilize is faster, as shown in the figure.

Large or long delay link, i.e. satellite or wireless link which has higher delay compared to normal wired links, for example 300 ms; higher slow-start threshold is needed for the throughput to become stable. In our experiments, flows for initial slow-start threshold value of 20 packets resulted in zero packet drop but the throughput fluctuates for about 20 seconds, i.e. from time 20 seconds to 40 seconds. This is an unstable period for the data transfer of audio or streaming media. The throughput then becomes stable 20 seconds after the flow started.

For the initial slow-start threshold value of 200 packets, the throughput can become stable faster. From the result shown, there is not much difference in terms of waiting time for the flow to become stable for initial slow-start threshold values of 100, 150 and 200 packets for VoIP G.711 data transmission.

There is no significant effect of variety of initial slow-start threshold value for faster or low delay link. A significant difference of the throughput of DCCP CCID2 flows can only be seen clearly using large delay link.

6. CONCLUSION

It is shown that UDP flow will definitely can not compromise when coexist with TCP flow. As an alternative to UDP, DCCP is a better transport protocol in delivering streaming data such as VoIP in which it can tolerate and friendlier with TCP.

From the simulation experiments, we can conclude that for the transmission of VoIP data with G.711 codec over large delay link, the suitable initial slow-start threshold value of 100 packets for both 20 ms and 300 ms link delay is acceptable for CCID2 congestion control. The use of very small slow-start threshold value will affect the performance of transmitting the voice data using DCCP CCID2 and resulted in longer time before the throughput become stable. As mentioned before, this will badly affect the perceived quality of voice delivered to the end users because during this unstable period, the throughput fluctuates very much and some of the throughput values are far below the VoIP G.711 encoding rate of 64 kbps.

On the other hand, the use of CCID3 for the transmission of VoIP G.711 data over large delay link is not very suitable due to its characteristics in which CCID3 suffers from lack of throughput over large delay link when sending data with small packet size.

REFERENCES

[1] *ns-2* network simulator, http://www.isi.edu/nsnam/ns/

[2] The *ns* Manual, http://www.isi.edu/nsnam/ns/ns-documentation.html

[3] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", RFC 4341, March 2006.

[4] S. Floyd and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.

[5] S. Floyd and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007.

[6] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[7] T. R. Henderson and R. H. Katz, "Transport Protocols for Internet-Compatible Satellite Networks", *IEEE Journal On Selected Areas In Communications*, Vol. 17, No. 2, pp. 326-344, February 1999.

[8] ISI-USC, "Transmission Control Protocol", RFC 793, 1981.

[9] V. Jacobson, "Congestion Avoidance and Control", In *Proc.* SIGCOMM, ACM, pp. 314-329, 1988.

[10] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[11] N.-E. Mattsson, "A DCCP module for *ns-2*", Masters's Thesis, Lulea University of Technology, 2004.

[12] J. Postel, "User Datagram Protocol", RFC 768, 1980.

[13] A. Sathiaseelan and G. Fairhurst, "Use of Quickstart for Improving the Performance of TFRC-SP Over Satellite Networks", In *Proc.* International Workshop on Satellite and Space Communications (IWSSC2006), Spain, pp. 46-50, 2006.